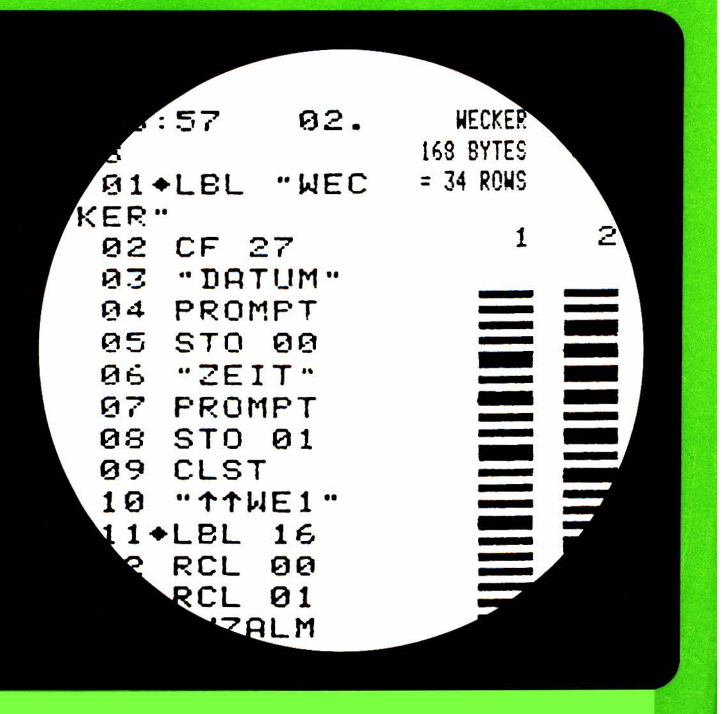
VIEWEG PROGRAMMBIBLIOTHEK
Mikrocomputer 28

HP-41 Hilfen und Anwendungen

Speichern, Sortieren, Mischen, Verwalten, Rechnen



Vieweg Programmbibliothek
Mikrocomputer 28

HP-41
Hilfen und Anwendungen

Aus dem Programm Mikrocomputer

Anwenderhandbuch HP-41 C/CV
von K. Gosmann

Softwareentwicklung am Beispiel einer Dateiverwaltung
von M. Gehret

Optimales Programmieren mit dem HP-41
von G. Kruse

Aus der Vieweg Programmbibliothek Mikrocomputer zum HP-41:

Band 14
Lineares Optimieren
11 HP-41-Programme

Band 15
Dienstprogramme (Tool-Kit)
für den HP-41

Band 18
Probleme der Festigkeitslehre
23 Programme für den HP-41

Band 21
HP-41 in der Praxis

Band 23
HP-41-Sammlung

Band 27
Kryptologie-Programme
(HP-41 C/CV)

Band 28
HP-41 – Hilfen und Anwendungen

Vieweg

Vieweg Programmbibliothek
Mikrocomputer Band 28

Harald Schumny (Hrsg.)

HP-41 – Hilfen und Anwendungen

Speichern, Sortieren, Mischen, Verwalten, Rechnen

14 Programme von Peter Reiter
mit Barcodes



Friedr. Vieweg & Sohn Braunschweig / Wiesbaden

CIP-Kurztitelaufnahme der Deutschen Bibliothek

Reiter, Peter:
HP 41 – Hilfen und Anwendungen: Speichern,
Sortieren, Mischen, Verwalten, Rechnen;
14 Programme; mit Barcodes / von Peter Reiter. –
Braunschweig; Wiesbaden: Vieweg, 1985.
(Vieweg-Programmbibliothek Mikrocomputer;
Bd. 28)
ISBN 3-528-04397-0

NE: GT

Der Autor des Bandes

Peter Reiter
Herrgottwiesgasse 202
A-8055 Graz

Das in diesem Buch enthaltene Programm-Material ist mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Der Autor übernimmt infolgedessen keine Verantwortung und wird keine daraus folgende oder sonstige Haftung übernehmen, die auf irgendeine Art aus der Benutzung dieses Programm-Materials oder Teilen davon entsteht.

1985

Umschlaggestaltung: Peter Lenz, Wiesbaden
Druck und buchbinderische Verarbeitung: W. Langelüddecke, Braunschweig
Printed in Germany

ISBN 3-528-04397-0

Inhaltsverzeichnis

Vorwort

In der heutigen Zeit – im Computerzeitalter – lassen sich Taschencomputer wie der HP-41CX gar nicht mehr wegdenken. Solche Kinder der Technik sind wahre Wunderwerke des Menschen und seines Denkens. Wenn man bedenkt, daß der HP-41CX vor ca. 20 Jahren so groß wie ein Einfamilienhaus gewesen wäre, kann man sich ungefähr ein Bild davon machen, welch gewaltiger Fortschritt in einem solchen elektronischen Gerät steckt.

Welchen Sinn hat aber ein „Taschenrechner" mit annähernd 250 Operationen im täglichen Leben?

Diese und ähnliche Fragen muß man sich selbst beantworten, denn jeder Mensch hat andere Bedürfnisse und keiner sieht in einem solchen Apparat das gleiche.

Was sehen die Benutzer, Programmierer und Techniker in einem solchen Gerät?

Sicherlich nicht das Wunderding, das jene Personen in ihnen sehen, die diesen „Dingen der Zukunft" aus dem Wege gehen, sondern ein unterstützendes und praktisches Hilfsmittel, mit dem sich außer den mathematischen Problemen noch eine Vielzahl anderer Aufgaben bewältigen lassen.

Freilich, überall sind Schranken gesetzt, so auch hier. Aber wenn man nicht an die Schranken herangeht, sie untersucht und sich mit ihnen beschäftigt, so wird man kaum jemals über sie hinwegkommen, sondern sie immer nur als Hindernis im Wege sehen.

Die Probleme die ich damit angesprochen haben möchte, lassen sich kurz erklären:
Die größte Schwierigkeit bei diesen kleinen Computern ist die Rechengeschwindigkeit. Sie läßt sich leider nicht mit dem minimalen Stromverbrauch in Einklang bringen; aber man muß ja irgendwo immer Abstriche machen. Hier möchte ich allerdings gleich erwähnen, daß der Hewlett-Packard-Rechner wohl einer der schnellsten unter den Rechnern in dieser Größen- und Leistungsklasse ist.

Irgendwann, meist nach der Zeit des Erlernens, wird sich dann das Problem des Speicherplatzes einstellen. HP hat sich schon vorher eingehend mit diesem Punkt beschäftigt und bietet neben den Speichererweiterungsmodulen auch ein Massenspeichermedium in Form eines Kassettenlaufwerks an. Zwar ist dies wohl nicht die beste Möglichkeit bezüglich Zugriffszeit, aber die Handhabung dieses Zusatzgerätes ist sehr einfach und die Bauweise äußerst robust.

Oft werden die Taschencomputer mit den diversen Mikrocomputern verglichen. Bei solchen Diskussionen hält man sich besser in seiner Meinungsäußerung zurück, außer man besitzt von beiden Klassen ein oder mehrere Exemplare. Probieren Sie es einmal aus und mischen Sie bei so einer Debatte mit; Sie werden feststellen was ich gemeint habe. Am Ende dieses Gesprächs werden Sie wissen, daß man Birnen und Zwetschgen kaum vergleichen kann.

Hinweise für die Programmierung

Leider bietet eine Schreibmaschine nicht die Möglichkeit, die gleichen Zeichen wie in der Anzeige des HP-41 darzustellen. Darum sollten Sie die folgenden Zeilen besonders gut durchlesen.

Programm-zeile	Befehl	Eingabe
XX	LBL "DIV"	(SHIFT) LBL – ALPHA – D – I – V – ALPHA
XX	APPEND " "	ALPHA – (SHIFT) K – SPACE – ALPHA (APPEND: ≙ im Alphamodus die Zweitfunktion der Taste "K")
XX	APPEND "="	ALPHA – (SHIFT) K – SPACE – = – ALPHA
XX	"OUTPUT"	ALPHA – O – U – T – P – U – T – ALPHA
XX	x	Das Multiplikationszeichen (*) wird im Buch durch ein kleines "x" ersetzt.

Die Ziffern, Zeichen oder Buchstaben die sich zwischen zwei Anführungszeichen befinden, sind in das ALPHA-Register einzugeben.

1 Datenspeicherung in einem File

Wie Sie sicherlich wissen werden, hat der HP-41CX max. 319 Datenspeicher. Da dies in manchen Fällen zu wenig sein kann, vor allem wenn man lange Programme abgespeichert hat, habe ich ein Programm zu Papier gebracht, mit dem Sie Daten in einem File abspeichern können, wobei auch eine Registerarithmetik möglich ist. Sollten Sie nun 2 Speichererweiterungsmodule in den Rechner stecken, so haben Sie 601 Datenregister zur Verfügung.

1.1 Programmbedienung

Bevor Sie die Speicherbefehle verwenden können, müssen Sie, sofern der File, den Sie benutzen wollen, schon vorhanden ist, den Filenamen im ALPHA-Register abspeichern und mit XEQ "FLREG" das Programm vorbereiten. Somit ist der File für die Datenspeicherung der momentane File. Sollte noch kein File für die "Registerarithmetik im erweiterten Speicher" vorhanden sein, so geben Sie im ALPHA-Register den Filenamen und im X-Stack die Anzahl der benötigten Register bekannt. Ebenso mit XEQ "FLREG" die Software starten. Nun ist der File für die Registerarithmetik vorhanden, außerdem ist dieser File der momentane File.
Die einzelnen Befehle können mittels einer XEQ-Anweisung bzw. Tastenzuordnung oder über das Label "FLAR" benützt

werden. Das Label "FLAR" ermöglicht es, in einer sich öfter wiederholenden Programmschleife verschiedene Fileregisteroperationen auszuführen, wobei die auszuführende Operation im ALPHA-Register bekanntgegeben wird.

1.2 Anweisungsliste

Prog. Zeile	Befehl	Kurzerklärung
Ø1	LBL "FLREG"	Dieser Programmteil dient dazu, einen
Ø2	SF 25	File anzulegen bzw., sollte dieser
Ø3	R↑	schon vorhanden sein, diesen zum
Ø4	CLX	momentanen File zu erklären.
Ø5	SEEKPTA	File anlegen (noch kein File vorhanden):
Ø6	FC?C 25	Filenamen = ALPHA-Register
Ø7	GTO 15	Registeranzahl = X-Stack
Ø8	RTN	File vorhanden:
Ø9	LBL 15	Filenamen = ALPHA-Register
1Ø	RDN	Jeweils Programmstart mit XEQ "FLREG".
11	CRFLD	
12	RTN	
13	LBL "FLAR"	Der Befehl "FLAR" ermöglicht ein
14	SF ØØ	indirektes Arbeiten mit den File-
15	R↑	operationen des Programms. Den zu
16	ASTO X	bearbeitenden Befehl speichern Sie im
17	GTO IND X	ALPHA-Register und starten, nachdem
		Sie die Daten je nach Operation ein-
		gegeben haben, mittels XEQ "FLAR".

Beispiel: 5Ø,4 ist zum Inhalt des File-
registers Nr. 1Ø zu addieren.

37	
38	
39	"FL+"
4Ø	1Ø
41	ENTER
42	5Ø,4
43	XEQ "FLAR"
44	
45	

18	LBL "FLSTO"	Abspeichern von Daten im File (gleicht
19	XEQ 22	dem Befehl STO).
2Ø	x<> y	Vor Programmstart:
21	SEEKPT	Fileregisternummer = Y-Stack
22	x<> y	Zahl = X-Stack
23	SAVEX	Nach Bearbeitung:
24	RTN	Fileregisternummer = Y-Stack
		Zahl = X-Stack
25	LBL "FLRCL"	Rückrufen von Daten aus dem File (gleicht
26	XEQ 22	dem Befehl RCL).
27	SEEKPT	Vor Programmstart:
28	GETX	Fileregisternummer = X-Stack
29	RTN	Nach Bearbeitung:
		Fileregisternummer = Y-Stack
		Zahl = X-Stack
3Ø	LBL "FL+"	Diese Befehle sind ähnlich den
31	LBL 16	Operationen ST+, ST-, STx und ST/,

```
32   XEQ 22
33   x<>y
34   SEEKPT
35   x<>y
36   GETX
37   FS?C Ø1
38   GTO 18
39   FS?C Ø2
4Ø   GTO 19
41   FS?C Ø3
42   GTO 2Ø
43   +
44   LBL 21
45   x<>y
46   SEEKPT
47   x<>y
48   SAVEX
49   RTN
5Ø   LBL "FL-"
51   SF Ø1
52   GTO 16
53   LBL "FLx"
54   SF Ø2
55   GTO 16
56   LBL "FL/"
57   SF Ø3
58   GTO 16
59   LBL 18
6Ø   x<>y
61   -
```

wobei hier nach der Berechnung das Ergebnis im X-Stack und im File enthalten ist.

Vor Programmstart:

Fileregisternummer = Y-Stack

Zahl = X-Stack

Nach Bearbeitung:

Fileregisternummer = Y-Stack

Ergebnis = X-Stack

```
GTO 21
LBL 19
x
GTO 21
LBL 20
x<> y
/
GTO 21
LBL 22
FC?C 00
RTN
CLX
RDN
RTN

LBL "FL<>RG"   (Nicht über "FLAR" möglich!)
XEQ 22         Dieser Programmteil tauscht einen
SEEKPT         Fileregisterinhalt mit dem Inhalt
GETX           eines Datenregisters.
x<> IND z      Vor Programmstart:
x<> y          Datenregisternummer = Y-Stack
SEEKPT         Fileregisternummer = X-Stack
x<>y           Nach Bearbeitung:
SAVEX          Inhalt vom Datenregister vor
RCL IND y      Bearbeitung = T-Stack
RDN            Inhalt vom Fileregister vor
RDN            Bearbeitung = Z-Stack
RTN            Datenregisternummer = Y-Stack
               Fileregisternummer = X-Stack
```

89	LBL "FLIND"	(Nicht über "FLAR" möglich!)
90	x<> y	Im Y-Stack geben Sie jenes File-
91	SEEKPT	register an, das die Parameter
92	x<> y	für den im ALPHA-Register spezifizierten
93	GETX	Befehl und im X-Stack gespeicherte
94	x<> y	Zahl aufweist.
95	GTO "FLAR"	Beispiel: "ST+" = ALPHA-Register

6 = Y-Stack

5 = X-Stack

Im Fileregister Nr. 6 = 4 und im Fileregister Nr. 4 = 10 enthalten.

Im Register Nr. 6 findet der Rechner die Parameterangabe 4, jetzt addiert er den Registerinhalt des Fileregisters Nr. 4 mit dem Inhalt des X-Stacks. Das Ergebnis wird im X-Stack bzw. Fileregister Nr. 4 abgespeichert.

Vor Programmstart:
Fileoperation = ALPHA-Register
Fileregister, das den Parameter enthält = Y-Stack
Zahl = X-Stack

Nach Bearbeitung:
Fileregister, das den Parameter enthalten hat = Z-Stack
Inhalt des mit dem Parameter spezifizierten Registers = Y-Stack
Ergebnis = X-Stack

```
96   LBL "FL<>FL"   (Nicht über "FLAR" möglich!)
97   XEQ 22         Austauschen zweier Fileregister.
98   SEEKPT         Vor Programmstart:
99   GETX           1. Fileregisternummer = Y-Stack
1ØØ  x<> z          2. Fileregisternummer = X-Stack
1Ø1  SEEKPT         Nach Bearbeitung:
1Ø2  GETX           Inhalt 1. Fileregister vor
1Ø3  x<> z          Bearbeitung = T-Stack
1Ø4  SEEKPT         Inhalt 2. Fileregister vor
1Ø5  x<> z          Bearbeitung = Z-Stack
1Ø6  SAVEX          1. Fileregisternummer = Y-Stack
1Ø7  x<> y          2. Fileregisternummer = X-Stack
1Ø8  SEEKPT
1Ø9  x<> t
11Ø  SAVEX
111  RDN
112  RDN
113  RTN

114  LBL "FLCLX"    Löschen von Registerbereichen.
115  XEQ 22         Vor Programmstart:
116  LBL ØØ         Das letzte zu löschende
117  SEEKPT         Register = Y-Stack
118  ENTER          das erste zu löschende
119  CLX            Register = X-Stack
12Ø  XEQ "FLSTO"
121  RDN            Den ganzen Inhalt des Files löschen
122  1              Sie mit dem Befehl "CLFL", der im
123  +              Benutzerhandbuch auf Seite 213
```

```
124  x<=y?        beschrieben ist.
125  GTO ØØ
126  CLST
127  RTN

128  END          Programmende
```

2 Sortieren von Zahlen

Diese Software verwendet als Subroutine Teile des Programms "FLREG"! Sollten Sie den Speicher mit zwei Memory-Modulen erweitert haben, so können Sie bis zu 601 Daten sortieren. Leider ist der Zeitaufwand sehr hoch.

2.1 Bedienungshinweise

Eingabe

Start mit XEQ "INPUT". Kurz darauf erscheint "HOW MANY?" in der Anzeige. Sie müssen dem Rechner nun bekanntgeben, wieviel Daten Sie sortieren wollen. Die Eingabe bestätigen Sie mit R/S. Als nächstes verlangt der Computer schon die Zahlen, wobei Sie jedesmal mit R/S abschließen.

Sortieren

Start mit XEQ "SORT", oder, wenn bei der Eingabe "END" in der Anzeige erscheint, mit R/S. Bei Beendigung des Programms ertönt ein Ton, und der Computer bleibt stehen.

Ausgabe

Die Daten rufen Sie mit "FLRCL" ab.

Beispiel: Sortieren von den folgenden 1Ø Zahlen:

37; 12; 8; 9; 41; 812; 7,2; 41Ø,2; 8,1; 81.

EINGABE	ANZEIGE
XEQ "INPUT"	"HOW MANY?"

EINGABE	ANZEIGE
1Ø R/S	Ø,ØØØØ
37 R/S	37,ØØØØ
12 R/S	12,ØØØØ
8 R/S	8,ØØØØ
9 R/S	9,ØØØØ
41 R/S	41,ØØØØ
812 R/S	812,ØØØØ
7,2 R/S	7,2ØØØ
41Ø,2 R/S	41Ø,2ØØØ
8,1 R/S	8,1ØØØ
81 R/S	81,ØØØØ
XEQ "SORT"	Ø,ØØØØ

Nach Beendigung des Sortiervorganges

EINGABE	ANZEIGE
Ø XEQ "FLRCL"	7,2ØØØ
1 XEQ "FLRCL"	8,ØØØØ
2 XEQ "FLRCL"	8,1ØØØ
3 XEQ "FLRCL"	9,ØØØØ
4 XEQ "FLRCL"	12,ØØØØ
5 XEQ "FLRCL"	37,ØØØØ
6 XEQ "FLRCL"	41,ØØØØ
7 XEQ "FLRCL"	81,ØØØØ
8 XEQ "FLRCL"	41Ø,2ØØØ
9 XEQ "FLRCL"	812,ØØØØ

2.2 Anweisungsliste

Prog. Zeile	Befehl	Prog. Zeile	Befehl
1	LBL "INPUT"	27	XEQ "FLRCL"
2	"HOW MANY?"	28	x <> y
3	PROMPT	29	RDN
4	"SORT"	3Ø	x < y?
5	SF 25	31	GTO 16
6	PRUFL	32	LBL 17
7	CF 25	33	CLA
8	XEQ "FLREG"	34	1
9	CLX	35	ST+ Ø1
1Ø	LBL ØØ	36	RDN
11	STOP	37	RCL Ø1
12	SF 25	38	SF 25
13	SAVEX	39	SEEKPT
14	FS?C 25	4Ø	RDN
15	GTO ØØ	41	FS?C 25
16	"END"	42	GTO 15
17	PROMPT	43	x <> y
18	LBL "SORT"	44	RCL ØØ
19	CLX	45	x<> y
2Ø	STO ØØ	46	XEQ "FLSTO"
21	1	47	1
22	STO Ø1	48	ST+ ØØ
23	LBL 15	49	RCL ØØ
24	RCL ØØ	5Ø	1
25	XEQ "FLRCL"	51	+
26	RCL Ø1	52	STO Ø1

Prog. Zeile	Befehl	Prog. Zeile	Befehl
53	SF 25	65	x<>y
54	SEEKPT	66	RDN
55	FS?C 25	67	x<>y
56	GTO 15	68	RCL ØØ
57	TONE 5	69	x<>y
58	CLST	7Ø	XEQ "FLSTO"
59	STOP	71	x<>y
6Ø	LBL 16	72	RDN
61	x<>y	73	x<>y
62	RCL Ø1	74	GTO 17
63	x<>y		
64	XEQ "FLSTO"	75	END

3 Fakultät bis 571 auf eine Stelle genau

3.1 Hintergrund

Stellen Sie sich vor, Sie mischen ein Kartenspiel mit 100 Karten. Was glauben Sie, wie viele Möglichkeiten gibt es, daß diese Spielkarten unterschiedlich aufeinander zu liegen kommen?
100, 1000 oder 100 000?
Es gibt wohl eine Unzahl mehr Konstellationen.
Um diese Möglichkeiten zu errechnen, können Sie nicht den Befehl "FACT" aus dem Rechner-ROM benutzen, da Ihnen der Computer mit "OUT OF RANGE" in der Anzeige stehenbleibt, außer Sie haben eine der Flags 24 bzw. 25 gesetzt. Um auch bei solchen Problemen noch ein Resultat zu erhalten und um eine höhere Genauigkeit zu erzielen, habe ich diese Software entwickelt.
Dieses Programm ermöglicht es, die Fakultät bis 571 auf eine Stelle genau zu errechnen. Es sei vor Rechenbeginn zu bedenken, daß bei Fakultät von 571 das Ergebnis nahezu 1330 Stellen aufweist. Außerdem sollten Sie sich nicht wundern, wenn Sie auf das Resultat etliche Stunden warten sollten, es geht eben nicht schneller. Sicherlich könnte man dieses Programm mit dem Worte "Spielerei" auf die Seite schieben. Dies sollte aber nicht Sinn und Zweck dieser Software sein, sondern ich wollte einfach zeigen,

wie man mit ein bißchen Kreativität die Grenzen von 1Ø Stellen leicht sprengen kann.

3.2 Programmbedienung und Beispiel

Die Zahl, von der die Fakultät berechnet werden soll, geben Sie in das X-Register und starten das Programm mit XEQ "xN". Wenn nun das Ergebnis ermittelt ist, stoppt der Computer und hat z. B. 63Ø = 4932178 in der Anzeige stehen, d.h., daß nach der "8" noch 63Ø Stellen folgen. Mit R/S werden Ihnen die nächsten 7 Stellen ausgegeben, Sie müssen diesen Vorgang so oft wiederholen, bis "END" in der Anzeige erscheint. Im Normalfall weist jede Ausgabe 7 Stellen auf, außer es sind führende Nullen vorhanden, z.B. Anzeige 77 = 342Ø6 eigentliches Ergebnis 77 = ØØ342Ø6.

Beispiel: n = 13

EINGABE	ANZEIGE
13 XEQ "xN"	7 = 622
R/S	Ø = 7Ø2Ø8ØØ
R/S	"END"

Verwendete Register: Ø bis 4 und 1Ø bis 199

3.3 Anweisungsliste

Prog. Zeile	Befehl	Kurzbeschreibung
1	LBL "xN"	Programmbezeichnung
2	INT	Kommastellen und "User Flag" löschen
3	CF 27	
4	572	Vergleichen, ob 572 kleiner oder
5	x<=y?	gleich der Eingabe (mit Eingabe ist
6	GTO 22	die Zahl, aus der die Fakultät
		berechnet werden soll, gemeint) ist.
		Sollte dieser Vergleich mit ja
		(do if true) beantwortet werden
		können, so wird zu Label 22 gesprungen.
7	CLX	2ØØ Register dem Datenspeicher
8	2ØØ	zuordnen.
9	PSIZE	
1Ø	CLRG	Register löschen.
11	RDN	Die Eingabe in Register Ø und 1Ø ab-
12	STO ØØ	legen. Das Register Ø enthält den
13	STO 1Ø	Multiplikationsfaktor, mit dem die
		Datenspeicher 1Ø bis 199 multipliziert
		werden.
14	1	1 vom Multiplikationsfaktor
15	ST- ØØ	(ehemalige Eingabe) subtrahieren.

10	10 in Register 1 speichern. Register
STO 01	1 dient als Datenspeicherzeiger für
	die indirekte Adressierung der
	Registerarithmetik.
LBL 16	Multiplikationsfaktor mit dem in
RCL 00	Register 1 definiertem Datenspeicher
STx IND 01	multiplizieren und den Überlauf
RCL 02	addieren. Ein Überlauf entsteht dann,
ST+ IND 01	wenn mehr als 7 Stellen in einem
	Datenspeicher (Reg. 10 - 199) sind.
1 E07	Den Überlauf errechnen, vom Daten-
ST/ IND 01	speicher subtrahieren und in Register
RCL IND 01	2 ablegen.
INT	
STO 02	
RCL IND 01	
FRC	
1 E07	
x	
STO IND 01	
LBL 17	Den Datenspeicher um 1 erhöhen und
1	abfragen ob der Zeiger auf Register
ST+ 01	200 weist, bei Beantwortung mit "JA"
RCL 01	zu Label 18 verzweigen.
200	
x = y?	
GTO 18	

40	RCL IND 01	Vergleichen, ob der nächste Daten-
41	x≠0?	speicher ungleich 0 ist, bei "JA",
42	GTO 16	Rücksprung zu Label 16.
43	RCL 02	Überlauf im leeren Datenspeicher ab-
44	STO IND 01	legen und Überlaufregister (Reg. 2)
45	CLX	löschen.
46	STO 02	
47	1	Abfrage, ob die nächsten 10 Daten-
48	ST+ 03	speicher leer sind, ist diese Abfrage
49	RCL 03	positiv, Rücksprung zu Label 17.
50	10	
51	x≠y?	
52	GTO 17	
53	LBL 18	Register 3 löschen.
54	CLX	
55	STO 03	
56	10	Datenspeicherzeiger auf 10
57	STO 01	zurücksetzen.
58	1	Multiplikationsfaktor um 1 verringern
59	ST- 00	und vergleichen, ob er ungleich 1 ist,
60	RCL 00	bei "JA", Rücksprung zu Label 16.
61	1	
62	x≠y?	
63	GTO 16	

64	RCLFLAG	Flags Ø bis 43 im Register 4 speichern
65	STO Ø4	(Hier beginnt der Programmteil für die Ausgabe des Ergebnisses.)
66	FIX Ø	Keine Kommastellen.
67	CF 29	Zifferntrennzeichenflag löschen.
68	199	Datenspeicherzeiger auf 199 setzen.
69	STO Ø1	
7Ø	LBL 23	Datenspeicher mit Ø vergleichen, bei "JA", Sprung zu Label 21. Wenn erster besetzter Datenspeicher gefunden ist, wird eine Tonkombination abgegeben.
71	RCL IND Ø1	
72	x=Ø?	
73	GTO 21	
74	BEEP	
75	LBL 24	Außer den angezeigten Stellen die noch vorhandene Stellenanzahl berechnen und ins Alpharegister bringen.
76	CLA	
77	RCL Ø1	
78	1Ø	
79	-	
8Ø	7	
81	x	
82	ARCL X	
83	APPEND " ="	An das Alpharegister "SPACE" und "=" anhängen.

84	ARCL IND Ø1	Inhalt des Datenspeichers an das
85	PROMPT	Alpharegister anhängen.
86	1	Datenspeicherzeiger um 1 verringern
87	ST- Ø1	und vergleichen, ob ungleich 9,
88	RCL Ø1	bei "JA", Rücksprung zu Label 24,
89	9	ansonst Flag Ø bis 43 mit dem in
9Ø	x≠y?	Register 4 enthaltenen Wert
91	GTO 24	restaurieren, Stack löschen, "END"
92	RCL Ø4	anzeigen und Programm stoppen.
93	STOFLAG	
94	CLST	
95	"END"	
96	PROMPT	
97	LBL 21	Datenspeicherzeiger um 1 verringern
98	1	und Rücksprung zu Label 23.
99	ST- Ø1	
1ØØ	GTO 23	
1Ø1	LBL 22	"ERROR" anzeigen und Programm stoppen.
1Ø2	"ERROR"	
1Ø3	PROMPT	
1Ø4	END	Programmende

4 Mischen von Ziffern, Zahlen oder Zeichen

4.1 Grundsätzliches

Nehmen Sie vier Blätter Papier und schreiben Sie je einen Buchstaben des Wortes "ROSE" auf ein Blatt. Jetzt vertauschen Sie die Blätter untereinander. Diese Buchstaben bilden nun maximal 24 (Fakultät aus 4 = 24) neue Wörter, wobei die meisten keine Bedeutung haben. Das gleiche läßt sich zum Beispiel mit Zahlen wiederholen.
Mit diesem Programm wird es Ihnen möglich, max. 1Ø Ziffern, Zahlen und Zeichen ohne großen Aufwand zu mischen. Leider ist der Zeitaufwand relativ hoch.

Beispiel: "ROSE" = ALPHA-Register
XEQ "MIS"

ROSE - ORSE - RSOE - SROE - OSRE - SORE - ROES - ORES -
REOS - EROS - OERS - EORS - RSEO - SREO - RESO - ERSO -
SERO - ESRO - OSER - SOER - OESR - EOSR - SEOR - ESOR -

Bei jeder neuen Stellung der Zeichen wird ein Ton ausgegeben und die neue Zeichenformation in die Anzeige gebracht. Die letzte Ausgabe wird durch die BEEP-Anweisung akustisch bekanntgegeben und bleibt daraufhin noch ca. 2 Sekunden sichtbar.

4.2 Anweisungsliste

Prog. Zeile	Befehl	Prog. Zeile	Befehl
1	LBL "MIS"	27	RCL 14
2	CLX	28	STx 13
3	x<>F	29	RDN
4	CF Ø8	3Ø	1
5	CF Ø9	31	-
6	,Ø15	32	ENTER↑
7	CLRGX	33	CLX
8	1Ø	34	x≠y?
9	STO 14	35	GTO Ø1
1Ø	ATOX	36	CLST
11	LBL ØØ	37	LBL Ø2
12	STO IND 1Ø	38	CLX
13	1	39	1
14	ST+ 11	4Ø	+
15	ST+ 1Ø	41	ST+ 12
16	CLST	42	RCL 14
17	ATOX	43	STx 12
18	x≠y?	44	RDN
19	GTO ØØ	45	RCL 11
2Ø	1	46	x≠y?
21	ST- 11	47	GTO Ø2
22	RCL 11	48	RCL 14
23	ENTER↑	49	ST/ 12
24	LBL Ø1	5Ø	LBL 18
25	RDN	51	RCL 11
26	ST+ 13	52	1

Prog. Zeile	Befehl
53	+
54	STO 1Ø
55	RCL 13
56	STO 15
57	LBL 15
58	RCL 15
59	RCL 14
6Ø	/
61	ENTER ↑
62	INT
63	STO 15
64	RDN
65	FRC
66	RCL 14
67	x
68	RCL 11
69	x< y?
7Ø	GTO 17
71	RDN
72	FS? IND X
73	GTO 17
74	SF IND X
75	RCL IND X
76	XTOA
77	DSE 1Ø
78	GTO 15
79	AVIEW
8Ø	TONE 5
81	CLA
82	CLX
83	x<> F
84	CF Ø8
85	CF Ø9
86	LBL 17
87	CLA
88	CLX
89	x<> F
9Ø	CF Ø8
91	CF Ø9
92	1
93	ST- 13
94	RCL 13
95	RCL 12
96	x< =y?
97	GTO 18
98	BEEP
99	CLST
1ØØ	CLX
1Ø1	PSE
1Ø2	PSE
1Ø3	CLD
1Ø4	STOP
1Ø5	END

verwendete Register: Ø bis 15

5 Division auf unendlich

Um herauszufinden, ob jede Division ein periodischer Dezimalbruch ist, hat ein Kollege in Zusammenarbeit mit mir dieses Programm entwickelt.

5.1 Programmbedienung

Nachdem der Dividend im Y-Register und der Divisor im X-Register ist, starten Sie das Programm mit XEQ "DIV". Zuerst wird der ganzzahlige Teil der Division ausgegeben, danach folgen die Kommastellen. Die Ergebnisse sind jeweils 15 Sekunden lang sichtbar. Ihre Länge umfaßt außer der ersten Ausgabe ausnahmslos 1Ø Stellen.
Um das Programm stoppen zu können, müssen Sie warten bis der Indikator für die Flag ØØ aufscheint, sobald Sie diese in der Anzeige sehen, betätigen Sie die Taste für den Dezimalpunkt.

Beispiel:	4ØØ1 ENTER	AUSGABE
	319	12
	XEQ "DIV"	5423197492
		163ØØ94Ø43
		8871473354
		2319749216

5.2 Anweisungsliste

Prog. Zeile	Befehl	Kurzbeschreibung
1	LBL "DIV"	Programmname
2	STO Ø2	Divisor im Register 2 und
3	RDN	Dividenden im Register 1 speichern.
4	STO Ø1	Register 3 und ALPHA-Register löschen.
5	CLX	
6	STO Ø3	
7	CLA	
8	RCLFLAG	Den Status der Flags Ø bis 43 im
9	STO ØØ	Register Ø speichern.
1Ø	FIX Ø	Keine Dezimalstellen anzeigen.
11	RCL Ø1	Division von Dividend und Divisor.
12	RCL Ø2	
13	/	
14	INT	Vom Ergebnis die Dezimalstellen löschen,
15	CF 29	Zifferntrennzeichenflag zurücksetzen
16	TONE 1	und einen tiefen Ton abgeben.
17	ARCL X	Den Wert im X-Stack in das ALPHA-
18	GTO 16	Register übertragen und zu Label 16 verzweigen.

19	LBL 15	Den Restwert der Division von
2Ø	RCL Ø1	Dividend und Divisor errechnen, mit
21	RCL Ø2	1Ø multiplizieren und als neuen
22	MOD	Dividenden im Register Ø abspeichern.
23	1Ø	
24	x	
25	STO Ø1	
26	RCL Ø2	Den neuen Dividenden mit dem Divisor,
27	/	dieser bleibt immer gleich, dividieren,
28	INT	die Kommastellen löschen und das
29	ARCL X	Ergebnis in das ALPHA-Register über-
		tragen.
3Ø	1	Sollten noch keine 1Ø Stellen er-
31	ST+ Ø3	rechnet sein, zurückspringen zu
32	RCL Ø3	Label 15.
33	1Ø	
34	x≠y?	
35	GTO 15	
36	CLX	Register 3 löschen und einen hohen
37	STO Ø3	Ton abgeben.
38	TONE 8	
39	LBL 16	Das ALPHA-Register in das Anzeige-
4Ø	AVIEW	Register übertragen und löschen.
41	CLA	

42	SF 00	Abfrage, ob die Dezimalpunkttaste
43	GETKEY	gedrückt wird, bei nein, Rücksprung
44	CF 00	zu Label 15.
45	83	
46	x≠y?	
47	GTO 15	
48	RCL 00	Restaurieren der Flags 0 bis 43 und
49	STOFLAG	löschen des Anzeige-, des ALPHA- und
50	CLD	der Stack-Register.
51	CLA	
52	CLST	
53	STOP	Programmstop
54	END	Programmende

6 Terminverwaltung

Durch die Gelegenheit, mittels genauer Zeit- und Datumsangaben ein Programm zu einer bestimmten Zeit zu starten, wird es möglich den Rechner als Terminkalender zu verwenden.

6.1 Die Software

Das Programm verwendet die Register Ø bis 6 und einen File im erweiterten Speicher. Die Software läßt sich in zwei große Teilprogramme unterteilen:
In "TERMIN" über diesem Label wird der Termin eingegeben und in "TERMIN?" mit diesem werden die Termine wieder abgerufen.

"TERMIN"

Wenn Sie das Programm mit XEQ "TERMIN" gestartet haben und noch kein File mit dem Namen "TERM1" im erweiterten Speicher vorhanden ist, erscheint kurz darauf "REGISTER?" in der Anzeige. Sie müssen nun bestimmen, wie groß der File für die Terminverwaltung sein soll. Die Eingabe beenden Sie mit R/S. Sollten Sie dem File mehr Register zuordnen wollen, als Speicherplatz im erweiterten Speicher frei ist, so erscheint "EXT MEM CLR" in der Anzeige. Sie

können die Anzeige mit "CLD" löschen. Nun steht im X-Stack jene Anzahl von Registern, die im erweiterten Speicher noch frei sind. Sie können, wenn Sie den vollen Speicherbereich zuteilen wollen, wieder R/S drücken, wobei "REGISTER?" nochmals in der Anzeige erscheint. Nun betätigen Sie abermals R/S und geben somit die Größe des Files bekannt. Als nächste Abfrage erscheint "TERM. DATE?" im Display. Nachdem Sie das Termindatum eingetastet haben, schließen Sie die Eingabe mit R/S ab, daraufhin erscheint "TERM. TIME?". Gleich wie beim Termindatum tätigen Sie die Eingabe der Terminzeit R/S. Sollte die Terminzeit später als 23 Uhr sein, so wird eine Sekunde lang "TOO LATE" angezeigt und zur Terminzeitabfrage zurückgesprungen. Mittels "TEXT" in der Anzeige erinnert der Computer, daß Sie auch eine Mitteilung bzw. Notiz abspeichern können. Geben Sie nun den Text ein (Achtung max. 24 Zeichen!) und schließen Sie mit R/S ab. Es erscheint die Frage "END J/N?". Betätigen Sie "N" für "NEIN", können Sie weiteren Text abspeichern. Diesen Vorgang können Sie so oft wiederholen, bis die Maximallänge eines Records erreicht ist. Bei "JA" (Drücken auf die Taste "J") stoppt der Rechner. Sollten Sie einen weiteren Termin zum abspeichern haben, so müssen Sie das Programm erneut mittels XEQ "TERMIN" starten.

Im erweiterten Speicher werden die Termine nicht geordnet, sondern nach ihrer Eingabe abgespeichert.

Jeden Tag um 23 Uhr 5Ø Minuten schaltet sich der Rechner ein und löscht sämtliche an diesem Tag aufgerufene Termine.

"TERMIN?"

Das Programm starten Sie mit XEQ "TERMIN?". Kurz darauf erscheint in der Anzeige "ALL? J/N",d.h., wollen Sie alle Termine oder nur jene, welche Sie mit einem Datum spezifizieren, abrufen? Sollten Sie die Taste "J" (ja) betätigen, so werden die Termine, so wie sie im File abgespeichert wurden, aufgerufen. Da im ALPHA-Register nur 24 Zeichen Platz haben, müssen Sie, nachdem Sie die 24 Zeichen gelesen haben, mit R/S abschließen. In der Anzeige erscheint darauf "LOOK/NEXT/END", wobei Sie, um weiteren Text zu bekommen, die Taste "L" für (LOOK) drücken müssen bzw. die Taste "N" für "NEXT", wenn Sie den nächsten Termin haben wollen. Um einen Programmstop zu bewirken, brauchen Sie nur die Taste "E" für "END" betätigen.
Sollten Sie nun beim Beginn (Abfrage "ALL? J/N") nichts drücken, "N" drücken oder eine andere Taste betätigen, so erscheint in der Anzeige "DATUM?". Ihre Datumseingabe beenden Sie mit R/S. Wird für diesen Tag ein Termin gefunden, so erscheint die Alarmzeit mit "XX,XX WHO J/N" in der Anzeige. D.h. wollen Sie wissen, wer, wen oder was Sie zu diesem Zeitpunkt besuchen müssen, so drücken Sie "J" für "JA" und "N" für "NEIN". Wenn Sie "N" betätigen, so wird der nächste Termin gesucht, sollte wiederum einer gefunden werden, wird er gleich angezeigt, wie der vorhergehende. Sie haben von neuem die Wahlmöglichkeit, sollte keiner gefunden werden stoppt der Rechner. Bei Betätigen der "J"-Taste erscheinen die ersten 24 Zeichen vom Record, da die Anzeige nur 24 Zeichen aufnimmt. Nachdem Sie den Text

gelesen haben, schließen Sie mit R/S ab. Nun erscheint die Abfrage "LOOK/NEXT/CLR". Sollten Sie mehr Text benötigen, drücken Sie "L" (LOOK). Wenn Sie den nächsten Termin sehen wollen, sofern einer vorhanden ist, drücken Sie die Taste "N" für "NEXT"; wenn der angezeigte Termin hinfällig ist, Sie ihn somit löschen wollen, betätigen Sie "C" für "CLEAR" Der Termin wird gelöscht, und sofern noch ein Termin für diesen Tag vorhanden ist, wird dieser angezeigt, wobei Sie gleich vorgehen wie bei den vorherigen Terminen. Sollten Sie keine der drei Tasten "L","N" oder "C" innerhalb von 1Ø Sekunden betätigen oder eine falsche, so erscheint wieder "LOOK/NEXT/CLR".

Terminerinnerung

Eine halbe Stunde vor jedem Termin meldet sich der Rechner mit 5 BEEP-Anweisungen und fragt, ob er den Termin vergessen (Taste "N"), oder ob er ihn mit dem Text anzeigen soll (Taste "J"). Sollte in den nächsten 1Ø Sekunden nach dem Alarm keine Taste betätigt werden, so meldet sich der Rechner nochmals 15 Minuten vor dem Termin. Bei der zweiten und daher letzten Terminerinnerung meldet sich der Computer mit 1Ø BEEP-Anweisungen und fragt, ob irgendeine Taste gedrückt wird, womit Sie ihm befehlen, den Termin auszugeben. Wird keine Taste betätigt, so wird der Termin automatisch gelöscht.

6.2 Anweisungsliste

Prog. Zeile	Befehl	Prog. Zeile	Befehl
1	LBL "TERMIN"	27	RDN
2	RCLFLAG	28	STO Ø1
3	STO Ø2	29	CLA
4	CLST	3Ø	ARCL X
5	SF 25	31	APPEND " "
6	"TERM1"	32	APPCHR
7	SEEKPTA	33	FIX 6
8	FC?C 25	34	CLST
9	XEQ 15	35	RCL ØØ
1Ø	FIX 6	36	RCL Ø1
11	"TERM. DATE?"	37	,3
12	PROMPT	38	HMS-
13	STO ØØ	39	"↑↑ TERM2"
14	CLA	4Ø	XYZALM
15	ARCL X	41	LBL 17
16	APPEND " "	42	"TEXT"
17	APPREC	43	AON
18	FIX 2	44	STOP
19	LBL 47	45	AOFF
2Ø	CLX	46	APPCHR
21	"TERM. TIME?"	47	LBL 18
22	PROMPT	48	"END J/N?"
23	RND	49	AVIEW
24	23	5Ø	GETKEY
25	x < y?	51	CLD
26	GTO 48	52	41

Prog. Zeile	Befehl
53	x=y?
54	GTO 17
55	RDN
56	25
57	x≠y?
58	GTO 18
59	CLD
6Ø	RCL Ø2
61	STOFLAG
62	CLA
63	CLST
64	STOP
65	LBL 15
66	"REGISTER?"
67	PROMPT
68	EMROOM
69	x < y?
7Ø	GTO 16
71	RDN
72	"TERM1"
73	CRFLAS
74	CLST
75	DATE
76	23,5
77	"⊦⊦TERM4"
78	XYZALM
79	RTN
8Ø	LBL 16
81	"EXT MEM CLR"
82	PROMPT
83	GTO 15
84	LBL "TERM2"
85	TIME
86	STO Ø1
87	RCLFLAG
88	STO Ø2
89	5
9Ø	STO ØØ
91	LBL 2Ø
92	BEEP
93	DSE ØØ
94	GTO 2Ø
95	"OUTP. J/N?"
96	AVIEW
97	GETKEY
98	25
99	x=y?
1ØØ	GTO 21
1Ø1	RDN
1Ø2	41
1Ø3	x=y?
1Ø4	OFF
1Ø5	CLD
1Ø6	CLST

Prog. Zeile	Befehl	Prog. Zeile	Befehl
1Ø7	DATE	134	OFF
1Ø8	RCL Ø1	135	FIX 2
1Ø9	FIX 2	136	RCL Ø1
11Ø	RND	137	RND
111	,15	138	,15
112	HMS+	139	HMS+
113	"⧫⧫TERM3"	14Ø	GTO 22
114	XYZALM	141	LBL 21
115	RCL Ø2	142	FIX 2
116	STOFLAG	143	RCL Ø1
117	CLST	144	RND
118	OFF	145	,3
119	LBL "TERM3"	146	HMS+
12Ø	TIME	147	LBL 22
121	STO Ø1	148	CLD
122	RCLFLAG	149	STO Ø1
123	STO Ø2	15Ø	"TERM1"
124	1Ø	151	CLST
125	STO ØØ	152	SEEKPTA
126	LBL 25	153	CLA
127	BEEP	154	FIX 6
128	DSE ØØ	155	DATE
129	GTO 25	156	ARCL X
13Ø	"TERMIN"	157	APPEND " "
131	AVIEW	158	FIX 2
132	GETKEY	159	RCL Ø1
133	x=Ø?	16Ø	ARCL X

Prog. Zeile	Befehl	Prog. Zeile	Befehl
161	POSFL	188	RCL ØØ
162	RCL Ø2	189	ARCL X
163	STOFLAG	19Ø	LBL 24
164	CLST	191	POSFL
165	RCLPT	192	-1
166	INT	193	x=y?
167	LBL 25	194	GTO 26
168	SF 25	195	RCLPT
169	GETREC	196	FRC
17Ø	RCLPT	197	x≠Ø?
171	INT	198	GTO 24
172	x≠y?	199	DELREC
173	GTO 5Ø	2ØØ	GTO 24
174	FC?C 25	2Ø1	LBL 26
175	GTO 5Ø	2Ø2	CLST
176	PROMPT	2Ø3	RCL ØØ
177	GTO 23	2Ø4	1
178	LBL "TERM4"	2Ø5	DATE+
179	DATE	2Ø6	23,5
18Ø	STO ØØ	2Ø7	"↑↑TERM4"
181	RCLFLAG	2Ø8	XYZALM
182	STO Ø2	2Ø9	RCL Ø2
183	"TERM1"	21Ø	STOFLAG
184	CLST	211	CLST
185	SEEKPTA	212	CLA
186	FIX 6	213	OFF
187	CLA	214	LBL "TERMIN?"

Prog. Zeile	Befehl	Prog. Zeile	Befehl
215	"ALL? J/N"	242	RCL ØØ
216	AVIEW	243	x=y?
217	GETKEY	244	GTO 3Ø
218	CLD	245	LBL 33
219	25	246	1
22Ø	x=y?	247	ST+ Ø1
221	GTO 39	248	GTO 31
222	RCLFLAG	249	LBL 3Ø
223	STO Ø5	25Ø	ATOX
224	"DATUM?"	251	ATOX
225	PROMPT	252	ASTO X
226	STO ØØ	253	RCL Ø2
227	1	254	x=y?
228	STO Ø1	255	GTO 38
229	"TERM2"	256	RDN
23Ø	ASTO Ø2	257	RCL Ø3
231	"TERM3"	258	x≠y?
232	ASTO Ø3	259	GTO 33
233	LBL 31	26Ø	,15
234	FIX 6	261	STO Ø6
235	RCL Ø1	262	LBL 36
236	SF 25	263	RCL Ø4
237	RCLALM	264	x<>y
238	FC?C 25	265	HMS+
239	GTO 35	266	STO Ø4
24Ø	STO Ø4	267	LBL 32
241	RDN	268	CLA

Prog. Zeile	Befehl	Prog. Zeile	Befehl
269	FIX 2	296	RCL Ø6
27Ø	ARCL Ø4	297	HMS+
271	APPEND " WHO J/N"	298	ARCL X
272	AVIEW	299	POSFL
273	GETKEY	3ØØ	RCLPT
274	CLD	3Ø1	INT
275	25	3Ø2	STO Ø4
276	x=y?	3Ø3	LBL 37
277	GTO 34	3Ø4	SF 25
278	RDN	3Ø5	GETREC
279	41	3Ø6	RCL Ø4
28Ø	x≠y?	3Ø7	RCLPT
281	GTO 32	3Ø8	INT
282	GTO 33	3Ø9	x≠y?
283	LBL 34	31Ø	GTO 49
284	FIX 6	311	FC?C 25
285	"TERM1"	312	GTO 49
286	CLST	313	PROMPT
287	SEEKPTA	314	LBL 46
288	RCL Ø1	315	"LOOK/NEXT/CLR"
289	RCLALM	316	AVIEW
29Ø	CLA	317	GETKEY
291	x <> y	318	CLD
292	ARCL X	319	33
293	APPEND " "	32Ø	x=y?
294	x <> y	321	GTO 37
295	FIX 2	322	RDN

Prog. Zeile	Befehl	Prog. Zeile	Befehl
323	41	35Ø	PROMPT
324	x=y?	351	"LOOK/NEXT/END"
325	GTO 33	352	AVIEW
326	RDN	353	GETKEY
327	13	354	CLD
328	x≠y?	355	CLA
329	GTO 46	356	33
33Ø	RCL Ø1	357	x=y?
331	CLALMX	358	GTO 4Ø
332	GTO 31	359	RDN
333	LBL 35	36Ø	41
334	RCL Ø5	361	x=y?
335	STOFLAG	362	GTO 41
336	CLST	363	RDN
337	CLA	364	15
338	STOP	365	x≠y?
339	LBL 38	366	GTO 39
34Ø	,3	367	CLST
341	STO Ø6	368	STOP
342	GTO 36	369	LBL 41
343	LBL 39	37Ø	1
344	"TERM1"	371	ST+ Ø2
345	CLST	372	RCL Ø2
346	SEEKPTA	373	SEEKPT
347	STO Ø2	374	GTO 4Ø
348	LBL 4Ø	375	LBL 48
349	GETREC	376	"TOO LATE"

Prog. Zeile	Befehl	Prog. Zeile	Befehl
377	AVIEW	386	GTO 33
378	PSE	387	LBL 5Ø
379	CLD	388	RCL Ø2
38Ø	GTO 47	389	STOFLAG
381	LBL 49	39Ø	CLST
382	"TEXTEND"	391	CLA
383	AVIEW	392	STOP
384	PSE	393	END
385	CLD		

7 Kuckucksuhr elektronisch

Der Vorteil des Rechners gegenüber der Kuckucksuhr liegt darin, daß man den "Kuckucksruf" bei Bedarf abstellen kann. Außerdem besteht die Möglichkeit, bei jeder vollen, jeder halben oder bei jeder viertel Stunde eine Tonkombination ertönen zu lassen.

7.1 Programmbedienung

Start mit XEQ "ZEITSIG".

Folgende Fragen müssen Sie dem Computer beantworten:

"ERSTES SIG"	Eingabe der Uhrzeit, wann der Rechner das erste Signal abgeben soll. (R/S)
"LETZTES SIG"	Eingabe jener Uhrzeit, bei der das letzte Signal abgegeben werden soll.
"1/4H? J/N"	Alle viertel Stunden ein Signal? "J" für ja und "N" für nein betätigen!
oder	
"1/2H? J/N"	Alle halben Stunden ein Signal? "J" für ja bzw. "N" für nein drücken.
"RTN/OFF?"	Nach dem Zeitsignal: soll der Rechner den Befehl "RTN" oder "OFF" ausführen? Jeweils den Anfangsbuchstaben für die gewünscht Anweisung betätigen.

Verwendete Flags: Ø = Zeitintervall 1/4h

1 = Zeitintervall 1/2h

2 = Ton für die 1/2h

Verwendete Flags: 3 = Ton für die 1/4h

4 = RTN / OFF

Verwendete Register: ØØ = erstes Zeitsignal

Ø1 = letztes Zeitsignal

Ø2 = Zeitintervall

Ø3 = Uhrzeit für Ton

Ø4 = Flags vom Zeitsignalprogramm

Ø5 = Flags vom Anwender

Ø6 = ersten 6 Alphazeichen

Ø7 = zweiten 6 Alphazeichen

Ø8 = dritten 6 Alphazeichen

Ø9 = vierten 6 Alphazeichen

1Ø = Inhalt vom T-Stack

11 = Inhalt vom Z-Stack

12 = Inhalt vom Y-Stack

13 = Inhalt vom X-Stack

14 = Inhalt vom Last X Register

15 = Zwischenspeicher (Arbeitsreg.)

LBL "ZE1" = für Stunden

LBL "ZE2" = für halbe Stunden

LBL "ZE3" = für viertel Stunden

7.2 Anweisungsliste

Prog. Zeile	Befehl	Prog. Zeile	Befehl
1	LBL "ZEITSIG"	26	LBL ØØ
2	CF 27	27	"1/4H? J/N"
3	SF 25	28	AVIEW
4	"♦♦ZE1"	29	GETKEY
5	CLALMA	3Ø	25
6	SF 25	31	x=y?
7	"♦♦ZE2"	32	GTO 16
8	CLALMA	33	RDN
9	SF 25	34	41
1Ø	"♦♦ZE3"	35	x≠y?
11	CLALMA	36	GTO ØØ
12	CF 25	37	LBL Ø1
13	CLST	38	"1/2H? J/N"
14	x<>F	39	AVIEW
15	,Ø14	4Ø	GETKEY
16	CLRGX	41	25
17	"ERSTES SIG"	42	x=y?
18	PROMPT	43	GTO 17
19	INT	44	RDN
2Ø	STO ØØ	45	41
21	STO Ø3	46	x≠y?
22	"LETZTES SIG"	47	GTO Ø1
23	PROMPT	48	LBL 18
24	INT	49	"RTN/OFF?"
25	STO Ø1	5Ø	AVIEW

Prog. Zeile	Befehl	Prog. Zeile	Befehl
51	GETKEY	78	x > y?
52	42	79	GTO Ø2
53	x=y?	8Ø	RCL Ø3
54	GTO 36	81	XEQ 25
55	RDN	82	XEQ 26
56	52	83	LBL 3Ø
57	x≠y?	84	CLST
58	GTO 18	85	x<> F
59	LBL 37	86	STO Ø4
6Ø	CLD	87	CLST
61	FS? ØØ	88	CLA
62	GTO 2Ø	89	STOP
63	FS? Ø1	9Ø	LBL 36
64	GTO 21	91	SF Ø4
65	1	92	GTO 37
66	LBL 22	93	LBL 43
67	STO Ø2	94	XEQ 32
68	TIME	95	OFF
69	RCL Ø1	96	LBL 16
7Ø	x < y?	97	SF ØØ
71	GTO 19	98	GTO 18
72	LBL Ø2	99	LBL 17
73	RCL Ø3	1ØØ	SF Ø1
74	RCL Ø2	1Ø1	GTO 18
75	HMS+	1Ø2	LBL 2Ø
76	STO Ø3	1Ø3	,15
77	TIME	1Ø4	GTO 22

Prog. Zeile	Befehl	Prog. Zeile	Befehl
1Ø5	LBL 21	132	RCL Ø3
1Ø6	,3	133	RCL Ø2
1Ø7	GTO 22	134	HMS+
1Ø8	LBL 19	135	STO Ø3
1Ø9	3Ø	136	RCL Ø1
11Ø	GTO 45	137	x < y?
111	LBL 41	138	GTO 41
112	42	139	RCL Ø3
113	LBL 45	14Ø	XEQ 25
114	STO 15	141	XEQ 26
115	CLST	142	LBL 42
116	DATE	143	FS? Ø4
117	1	144	GTO 43
118	DATE+	145	XEQ 32
119	RCL ØØ	146	RTN
12Ø	STO Ø3	147	LBL "ZE2"
121	"⊦⊦ZE1"	148	TONE 8
122	XYZALM	149	TONE 4
123	GTO IND 15	15Ø	TONE 8
124	LBL "ZE1"	151	GTO 4Ø
125	TONE 5	152	LBL "ZE3"
126	TONE 6	153	TONE 7
127	TONE 7	154	TONE 3
128	TONE 8	155	GTO 4Ø
129	TONE 3	156	LBL 32
13Ø	LBL 4Ø	157	CLA
131	XEQ 31	158	ARCL Ø6

Prog. Zeile	Befehl	Prog. Zeile	Befehl
159	ARCL Ø7	186	ASTO Ø7
16Ø	ARCL Ø8	187	ASHF
161	ARCL Ø9	188	ASTO Ø8
162	CLST	189	ASHF
163	x<> F	19Ø	ASTO Ø9
164	STO Ø4	191	CLA
165	RCL Ø5	192	RCLFLAG
166	STOFLAG	193	STO Ø5
167	RCL 14	194	RCL Ø4
168	STO L	195	x<> F
169	RCL 1Ø	196	CLST
17Ø	RCL 11	197	RTN
171	RCL 12	198	LBL 25
172	RCL 13	199	FRC
173	RTN	2ØØ	,3
174	LBL 31	2Ø1	x=y?
175	STO 13	2Ø2	GTO 23
176	RCL L	2Ø3	RDN
177	STO 14	2Ø4	,15
178	RDN	2Ø5	x=y?
179	STO 12	2Ø6	GTO 24
18Ø	RDN	2Ø7	RDN
181	STO 11	2Ø8	,45
182	RDN	2Ø9	x=y?
183	STO 1Ø	21Ø	GTO 24
184	ASTO Ø6	211	CF Ø2
185	ASHF	212	CF Ø3

Prog. Zeile	Befehl	Prog. Zeile	Befehl
213	CLST	227	GTO 28
214	RTN	228	"⊢ZE1"
215	LBL 23	229	LBL 29
216	SF Ø2	23Ø	CLST
217	CF Ø3	231	RCL Ø3
218	RTN	232	XYZALM
219	LBL 24	233	RTN
22Ø	SF Ø3	234	LBL 27
221	CF Ø2	235	"⊢ZE2"
222	RTN	236	GTO 29
223	LBL 26	237	LBL 28
224	FS? Ø2	238	"⊢ZE3"
225	GTO 27	239	GTO 29
226	FS? Ø3	24Ø	END

8 Wecken durch den HP-41CX

Eine der vielen Möglichkeiten, sich aus dem Schlummer reißen zu lassen, ist jene, den Wiederholungsintervall des Alarms immer kürzer zu setzen. Genau das wird von diesem Programm gemacht und zwar im Abstand von 4, 3, 2 und 1 Minute.

8.1 Programmbedienung

Start mit XEQ "WECKER". Die Eingabe vom "DATUM" und "ZEIT" mit R/S abschließen.
Um den Wecker abzustellen, betätigen Sie beim "Läuten" die R/S- oder die ON/OFF-Taste, und Sie werden keinen Ton von Ihrem Rechner mehr hören. Sollten Sie hingegen sich nicht um die Piepserei kümmern und keine Taste drücken, so meldet sich der Computer bis zu viermal wieder.

Verwendete Register: Ø bis 2

8.2 Anweisungsliste

Prog. Zeile	Befehl
1	LBL "WECKER"
2	CF 27
3	"DATUM"
4	PROMPT
5	STO ØØ
6	"ZEIT"
7	PROMPT
8	STO Ø1
9	CLST
1Ø	"⊦WE1"
11	LBL 16
12	RCL ØØ
13	RCL Ø1
14	XYZALM
15	CLST
16	CLA
17	RTN
18	LBL "WE1"
19	,Ø4
2Ø	"⊦WE2"
21	LBL 99
22	STO Ø2
23	15
24	LBL ØØ
25	TONE 8
26	TONE 1
27	DSE X
28	GTO ØØ
29	TIME
3Ø	1ØØ
31	x
32	INT
33	1ØØ
34	/
35	RCL Ø2
36	HMS+
37	STO Ø1
38	CLST
39	XEQ 16
4Ø	OFF
41	LBL "WE2"
42	,Ø3
43	"⊦WE3"
44	GTO 99
45	LBL "WE3"
46	,Ø2
47	"⊦WE4"
48	GTO 99
49	LBL "WE4"
5Ø	,Ø1
51	"⊦WE5"
52	GTO 99
53	LBL "WE5"
54	15
55	LBL Ø1
56	BEEP
57	DSE X
58	GTO Ø1
59	OFF
6Ø	END

9 Löschen des erweiterten Speichers

Um einen File aus dem erweiterten Speicher zu entfernen, mußte man ihn bisher im ALPHA-Register definieren und mit dem Befehl "PURFL" löschen. Dies kann, wenn man sehr viele Files hat, zu einer zeitraubenden Tätigkeit werden.
Die beiden Programme "CLEMX" (clear extended memory by X) und "CLEM" (clear extended memory) helfen Ihnen, diesen Aufwand zu minimieren.
"CLEM" löscht den gesamten erweiterten Speicher, und bei "CLEMX" geben Sie mit einer Zahl im X-Stack bekannt, wie groß der Speicherplatz ist, den Sie benötigen.

9.1 Programmbeschreibung

Die beiden Programme sind als Subroutinen ausgelegt, d.h. sollten Sie "CLEM" oder "CLEMX" mit einer XEQ-Anweisung anspringen, so wird das Programm ausgeführt, bei Ende desselben zum nächstenfolgenden Befehl nach XEQ "?" zurückgesprungen und das Hauptprogramm weiterbearbeitet. Sollten Sie hingegen im Rechenmodus die Anweisung XEQ "CLEM" bzw. XEQ "CLEMX" geben, so wird der RTN-Befehl zum Schluß als "STOP" interpretiert.
Bei "CLEMX" steht nach der Beendigung des Programms im X-Stack, wieviele Register Sie nun wirklich zur Verfügung haben, da das Programm von vorne nach hinten, somit in der

Reihenfolge wie bei dem Befehl "CAT 4" ausgegeben, die Files aus dem erweiterten Speicher löscht und vergleicht, ob der erforderliche Platz schon frei ist.

9.2 Anweisungsliste

Prog. Zeile	Befehl	Kurzbeschreibung
1	LBL "CLEM"	Programmname
2	CLST	Stack löschen.
3	1	Erstes File vom externen Speicher in
4	EMDIRX	das X- und ALPHA-Register übertragen.
5	x=y?	Wenn kein File mehr vorhanden ist,
6	RTN	zurückspringen zu vorher benützter Unterprogramm- oder Programmebene bzw. Programmstop
7	CLX	File aus dem erweiterten Speicher
8	PURFL	löschen. Rücksprung zu Programm-
9	GTO "CLEM"	anfang.
1Ø	END	Programmende

Prog. Zeile	Befehl	Kurzbeschreibung
1	LBL "CLEMX"	Programmname
2	EMROOM	Ist der Paum im erweiterten Speicher
3	x > y?	größer oder gleich der Eingabe? Wenn
4	RTN	ja, Rücksprung zu vorheriger
5	x=y?	Unterprogamm- oder Programmebene
6	RTN	bzw. Programmstop.
7	RDN	Erstes File vom erweiterten Speicher
8	1	in das X- und ALPHA-Register über-
9	EMDIRX	tragen.
10	CLX	File aus dem erweiterten Speicher
11	SF 25	löschen. Sollte kein File mehr im
12	PURFL	erweiterten Speicher sein, Rücksprung
13	RDN	zu vorheriger Unterprogramm- oder
14	EMROOM	Programmebene bzw. Programmstop.
15	FC?C 25	
16	RTN	
17	RDN	Rücksprung zu Programmanfang.
18	GTO "CLEMX"	
19	END	Programmende

10 Primfaktorzerlegung

Dieses Programm zerlegt jede Zahl in ihre Primzahlen.

10.1 Programmbedienung

Die zu zerlegende Zahl in das X-Register eingeben und das Programm mit XEQ "PRIM" starten.
Jeder Primfaktor außer dem letzten wird 1 Sekunde lang angezeigt.

Verwendete Register: 16 bis 19

Beispiel: 39240 XEQ "PRIM"

AUSGABE
2
2
2
3
5
109

10.2 Anweisungsliste

Prog. Zeile	Befehl	Kurzbeschreibung
1	LBL PRIM"	Programmname
2	CF 27	USERS-Flag löschen und Eingabe in
3	STO 16	Register 16 speichern.
4	LBL Ø3	Von der Zahl im X-Stack die Quadrat-
5	SQRT	wurzel ziehen, die Kommastellen
6	INT	löschen und das Ergebnis im
7	STO 19	Register 19 speichern.
8	FS? ØØ	Wurde gerade ein Primfaktor ausgegeben?
9	GTO Ø2	Wenn ja, Sprung zu Label Ø2, bei nein
1Ø	1	1 in Register und 2 in Register 17
11	STO 18	speichern. (Reg. 17 = Divisionsfaktor)
12	2	
13	STO 17	
14	LBL Ø2	Flag Ø löschen.
15	CF ØØ	
16	LBL ØØ	Ist der Divisionsfaktor größer als
17	RCL 16	die Wurzel aus der Eingabe? Wenn ja,
18	RCL 17	Sprung zu Label 15, ansonst Division
19	RCL 19	von Eingabe mit Divisionsfaktor.

2Ø	x<y?	Ganzzahligen Teil, des Ergebnisses
21	GTO 15	löschen. Sind Kommastellen vorhanden?
22	RDN	Wenn ja, Sprung zu Label Ø1, sonst
23	/	Divisionsfaktor um 1 erhöhen und zu
24	FRC	Label ØØ zurückspringen.
25	x=Ø?	
26	GTO Ø1	
27	RCL 18	
28	ST+ 17	
29	GTO ØØ	
3Ø	LBL Ø1	Primfaktor 1 sec. lang anzeigen,
31	RCL 17	Eingabe mit Primfaktor dividieren,
32	PSE	Ergebnis in den X-Stack bringen
33	RCL 17	und im Register 16 speichern. Flag ØØ
34	ST/16	setzen und Rücksprung zu Label Ø3.
35	SF ØØ	
36	RCL 16	
37	GTO Ø3	
38	LBL 15	Flag ØØ löschen, letzten Primfaktor
39	CF ØØ	ins X-Register übertragen und
4Ø	RCL 16	Programmstop.
41	RTN	
42	END	Programmende

11 Quersummenberechnung

11.1 Grundsätzliches

Für ein paar mathematische Tests habe ich dieses Programm zu Papier gebracht.
Wenn Sie eine beliebige Zahl, z.B. 179, notieren, so haben Sie die Zahlen 1, 7 und 9 zu Papier gebracht. Diese Zahlen addiert der Rechner. Bei unserem Beispiel hätten wir als Ergebnis 17 erhalten. Die Maximallänge ist die Stellenzahl der Mantisse. Sollte eine Zahl Kommastellen aufweisen, so wird das Komma so lange nach rechts verschoben, bis kein Dezimalbruch mehr vorhanden ist.

Dateneingabe

Diejenige Zahl, von der Sie die Quersumme errechnet haben wollen, geben Sie in den X-Stack und starten das Programm mit XEQ "QSUM". Sobald der Computer das Ergebnis gefunden hat, bleibt er mit dem Ergebnis in der Anzeige stehen. Das Erwähnenswerte bei dieser Software ist wohl, daß keine Datenspeicher zur Datenzwischenspeicherung verwendet werden, sondern nur mit den Stack-Registern gearbeitet wird.

11.2 Anweisungsliste

Prog. Zeile	Befehl	Kurzbeschreibung
1	LBL "QSUM"	Programmname
2	LBL 01	Sollte eine Zahl Kommastellen auf-
3	ENTER	weisen, so wird diese so lange mit
4	FRC	1Ø multipliziert, bis kein Dezimal-
5	x=Ø?	bruch mehr vorhanden ist.
6	GTO ØØ	
7	CLX	
8	1Ø	
9	x	
1Ø	GTO Ø1	
11	LBL ØØ	Löschen des X- und Z-Stacks und ein
12	CLX	Verschieben der Stack-Inhalte nach
13	STO z	unten.
14	RDN	
15	LBL Ø3	Eingabe mit 1Ø dividieren und im
16	1Ø	Z-Stack speichern.
17	/	
18	STO z	
19	FRC	Den ganzzahligen Anteil der im X-Stack
2Ø	1Ø	enthaltenen Zahl löschen, den Dezimal-
21	x	bruch mit 1Ø multiplizieren und den
22	ST+ y	Y-Stack mit der Zahl addieren.

23	CLX	X-Stack löschen. (Mit diesem Befehl wird auch der Stack-Lift gesperrt.)
24	RCL z	Den Z-Stackinhalt in den X-Stack über-
25	INT	tragen, die Kommastellen löschen und
26	x≠Ø?	vergleichen, ob die im X-Stack
27	GTO Ø3	enthaltene Zahl ungleich Ø ist. Bei
28	RCL y	positiver Beantwortung der Abfrage,
29	RTN	Rücksprung zu Label Ø3, sonst Ausgabe der Quersumme.
3Ø	END	Programmende

12 Fakultäten, Variationen und Kombinationen auch für Mengen größer 69

12.1 Grundsätzliches, Beispiele

Berechnung der Anzahl der möglichen Kombinationen und Variationen von n Objekten, wenn r Objekte gleichzeitig entnommen werden. Dazu Berechnung der Fakultät von positiven ganzen Zahlen.

Dieses Programm führt drei Funktionen aus.

Fakultät: Berechnung der Fakultät (n!) einer positiven ganzen Zahl.

n! = (n) (n - 1) (n - 2)   (n - (n-1))

Variationen: Berechnung der Anzahl der möglichen Variationen von n Objekten, wenn r Objekte gleichzeitig entnommen werden.

P = n! / (n - r)!

Kombinationen: Berechnung der Anzahl der möglichen Kombinationen von n Objekten, wenn r Objekte gleichzeitig entnommen werden.

C = n! / r! (n - r)!

Beispiel für Fakultät:

Die einzelnen Bücher eines zwanzigbändigen Lexikons werden in zufälliger Reihenfolge im Regal aneinandergestellt. Wieviele mögliche Anordnungen gibt es?

EINGABE	ANZEIGE
Start mit	
XEQ "FAK"	N?
2Ø R/S	Anzeige der Uhrzeit und nach einem Piepston das Ergebnis, in unserem Fall 2,4329Ø2 18

Es gibt 2,4329Ø2 . $1Ø^{18}$ mögliche Anordnungen.

Beispiel für Variationen:

37 Studenten legen eine Prüfung für die Bewilligung eines Stipendiums ab. Die Studenten mit den 4 besten Prüfungsergebnissen erhalten Stipendien von S 4ØØØØ, S 3ØØØØ, S 2ØØØØ bzw. S 1ØØØØ. Wieviele unterschiedliche Resultate sind möglich? (S = Schilling)

EINGABE	ANZEIGE
Start mit	
XEQ "VAR"	N?
37 R/S	R?
4 R/S	Anzeige der Uhrzeit und nach einem Piepston das Ergebnis, in unserem Fall 1585Ø8Ø

Beispiel für Kombinationen:

Beim Kartengeben erhält ein Spieler 4 Karten aus einem 11Ø-iger Blatt. Wieviele mögliche Hände kann er bekommen?

EINGABE	ANZEIGE
Start mit	
XEQ "KOM"	N?
11Ø R/S	R?
4 R/S	Anzeige der Uhrzeit und nach einem Piepston das Ergebnis, in unserem Fall 5773187

Sollte zum Beispiel die Fakultät von n=7Ø errechnet werden, so wird das Ergebnis wie folgend ausgegeben:

MAN= 1,197857 R/S drücken

EXP= 1ØØ

n! = 1,197857 . $1Ø^{1ØØ}$

12.2 Anweisungsliste

Prog. Zeile	Befehl	Prog. Zeile	Befehl
1	LBL "FAK"	7	GTO 18
2	CF 25	8	LBL "VAR"
3	XEQ 16	9	CF 27
4	XEQ 21	1Ø	XEQ 16
5	RCL ØØ	11	XEQ 17
6	XEQ 15	12	XEQ 21

Prog. Zeile	Befehl
13	XEQ 19
14	RCL ØØ
15	XEQ 15
16	RCL Ø4
17	-
18	GTO 18
19	LBL "KOM"
2Ø	CF 27
21	XEQ 16
22	XEQ 17
23	XEQ 21
24	XEQ 19
25	RCL Ø1
26	XEQ 15
27	ST+ Ø4
28	RCL ØØ
29	XEQ 15
3Ø	RCL Ø4
31	-
32	GTO 18
33	LBL 16
34	"N?"
35	PROMPT
36	ABS
37	INT
38	STO ØØ
39	RTN
4Ø	LBL 17
41	"R?"
42	PROMPT
43	ABS
44	INT
45	STO Ø1
46	RCL ØØ
47	x > y?
48	RTN
49	"N > R???"
5Ø	PROMPT
51	LBL 15
52	STO Ø2
53	CLX
54	STO Ø3
55	LBL ØØ
56	RCL Ø2
57	LOG
58	ST+ Ø3
59	DSE Ø2
6Ø	GTO ØØ
61	RCL Ø3
62	RTN
63	LBL 18
64	TONE 5
65	STO Ø2
66	INT

Prog. Zeile	Befehl	Prog. Zeile	Befehl
67	SF 25	87	RCL Ø2
68	1Ø↑X	88	INT
69	FC?C 25	89	"EXP= "
7Ø	GTO Ø1	9Ø	ARCL X
71	RCL Ø2	91	AVIEW
72	FRC	92	RTN
73	1Ø↑X	93	LBL 19
74	x	94	RCL ØØ
75	,5	95	RCL Ø1
76	+	96	-
77	INT	97	XEQ 15
78	CLD	98	STO Ø4
79	STOP	99	RTN
8Ø	LBL Ø1	1ØØ	LBL 21
81	RCL Ø2	1Ø1	CLA
82	FRC	1Ø2	TIME
83	1Ø↑X	1Ø3	ATIME
84	"MAN= "	1Ø4	AVIEW
85	ARCL X	1Ø5	RTN
86	PROMPT	1Ø6	END

13 Bruchrechnen

13.1 Vorbemerkungen, Beispiele

Es wird wohl ein jeder mit der Achsel zucken, wenn Sie ihn fragen, wie das Ergebnis von $\frac{5}{7} : \frac{1}{10}$ lautet.

Sie wissen natürlich in Kürze, sofern Sie richtig getippt haben, daß das Resultat nur $\frac{50}{7}$ bzw. $7\,\frac{1}{7}$ sein kann.

Eine jede Dezimalzahl läßt sich als Bruch darstellen, man muß nur wissen wie.

z.B.: $50{,}14 = 50\,\frac{14}{100}$ oder

$\frac{5014}{100}$ oder

$\frac{2507}{50}$

$17 = \frac{17}{1}$

$3{,}333 = 3\,\frac{333}{1000}$

Wie heißen die Zahlen bei einem Bruch?

Ganze Zahl —— $17\,\frac{1}{9}$ (1: Zähler, 9: Nenner)

Wie können Sie nun mit diesem Programm Bruchrechnen?

Anhand von Beispielen sei dies erklärt !

$$14\,\frac{3}{7} : \frac{1}{6} =$$

EINGABE	ANZEIGE
Start mit	
XEQ "BRE1"	
(BRE ist die	
Abkürzung für	
Bruchrechnen)	GANZE ZAHL=
14 R/S	ZAEHLER=
3 R/S	NENNER=
7 R/S	GANZE ZAHL=
R/S	ZAEHLER=
1 R/S	NENNER=
6 R/S	BEFEHL?
XEQ "B/"	ZAEHLER= 6Ø6
R/S	NENNER= 7
R/S	GANZE ZAHL= 86
R/S	ZAEHLER= 4
R/S	NENNER= 7

Ergebnis $= \frac{606}{7} = 86\,\frac{4}{7}$

Neues Beispiel:

$$1\,\frac{7}{69} \cdot 13\,\frac{7}{9} - \frac{18}{111} =$$

(Achtung: Punkt- vor Strichrechnung)

EINGABE	ANZEIGE
XEQ "BRE1"	GANZE ZAHL=
1 R/S	ZAEHLER=
7 R/S	NENNER=
69 R/S	GANZE ZAHL=
13 R/S	ZAEHLER=
7 R/S	NENNER=
9 R/S	BEFEHL?
XEQ "Bx"	ZAEHLER= 9424
XEQ "BRE2"	GANZE ZAHL=
R/S	ZAEHLER=
18 R/S	NENNER=
111 R/S	BEFEHL?
XEQ "B-"	ZAEHLER= 344962
R/S	NENNER= 22977
R/S	GANZE ZAHL= 15
R/S	ZAEHLER= 3Ø7
R/S	NENNER= 22977

Ergebnis = $\frac{344962}{22977} = 15 \frac{307}{22977}$

ALPHA-Labels und ihre Bedeutung

LBL "BRE1" = Programmstart

LBL "BRE2" = dient zur Eingabe, wenn mehrere Rechenvorgänge bei einer Rechnung erforderlich sind.

LBL "B+" = addieren
LBL "B-" = subtrahieren
LBL "Bx" = multiplizieren
LBL "B/" = dividieren
} der jeweiligen Brüche

13.2 Anweisungsliste

Prog. Zeile	Befehl	Prog. Zeile	Befehl
1	LBL "BRE1"	26	ABS
2	CLST	27	RCL Ø2
3	XEQ 15	28	x
4	PROMPT	29	ST+ Ø1
5	STO ØØ	3Ø	FS?C ØØ
6	XEQ 16	31	XEQ Ø5
7	PROMPT	32	RCL Ø3
8	STO Ø1	33	x < Ø?
9	XEQ 17	34	SF ØØ
1Ø	PROMPT	35	ABS
11	STO Ø2	36	RCL Ø5
12	LBL "BRE2"	37	x
13	CLST	38	ST+ Ø4
14	XEQ 15	39	FS?C ØØ
15	PROMPT	4Ø	XEQ Ø6
16	STO Ø3	41	CLST
17	XEQ 16	42	STO ØØ
18	PROMPT	43	STO Ø3
19	STO Ø4	44	"BEFEHL?"
2Ø	XEQ 17	45	PROMPT
21	PROMPT	46	LBL Ø5
22	STO Ø5	47	-1
23	RCL ØØ	48	STx Ø1
24	x < Ø?	49	RTN
25	SF ØØ	5Ø	LBL Ø6

Prog. Zeile	Befehl	Prog. Zeile	Befehl
51	-1	78	STx Ø2
52	STx Ø4	79	LBL 19
53	RTN	8Ø	CF ØØ
54	LBL "B+"	81	RCL Ø1
55	XEQ 18	82	x < Ø?
56	RCL Ø1	83	XEQ 2Ø
57	RCL Ø4	84	STO Ø3
58	+	85	RCL Ø2
59	STO Ø1	86	STO Ø4
6Ø	GTO 19	87	LBL ØØ
61	LBL "B-"	88	CLST
62	XEQ 18	89	RCL Ø2
63	RCL Ø1	9Ø	x=y?
64	RCL Ø4	91	GTO Ø1
65	-	92	RCL Ø1
66	STO Ø1	93	RCL Ø2
67	GTO 19	94	/
68	LBL "Bx"	95	FRC
69	RCL Ø4	96	RCL Ø2
7Ø	STx Ø1	97	x
71	RCL Ø5	98	,1
72	STx Ø2	99	+
73	GTO 19	1ØØ	INT
74	LBL "B/"	1Ø1	x <> Ø2
75	RCL Ø5	1Ø2	STO Ø1
76	STx Ø1	1Ø3	GTO ØØ
77	RCL Ø4	1Ø4	LBL Ø1

Prog. Zeile	Befehl	Prog. Zeile	Befehl
1Ø5	RCL Ø1	132	STO ØØ
1Ø6	STO Ø2	133	FRC
1Ø7	RCL Ø3	134	RCL Ø2
1Ø8	x<>y	135	x
1Ø9	/	136	RCLFLAG
11Ø	STO Ø1	137	x<>y
111	RCL Ø4	138	FIX Ø
112	RCL Ø2	139	RND
113	/	14Ø	x<>y
114	STO Ø2	141	STOFLAG
115	LBL Ø2	142	x<>y
116	RCL Ø1	143	STO Ø1
117	LBL Ø3	144	RCL ØØ
118	FS?C ØØ	145	INT
119	CHS	146	STO ØØ
12Ø	STO Ø1	147	LBL Ø4
121	XEQ 16	148	RCL ØØ
122	XEQ 25	149	XEQ 15
123	PROMPT	15Ø	XEQ 25
124	RCL Ø2	151	PROMPT
125	XEQ 17	152	RCL Ø1
126	XEQ 25	153	ABS
127	AVIEW	154	XEQ Ø3
128	RTN	155	STOP
129	RCL Ø1	156	GTO Ø4
13Ø	RCL Ø2	157	LBL 18
131	/	158	RCL Ø5

Prog. Zeile	Befehl
159	RCL Ø2
16Ø	STx Ø4
161	STx Ø5
162	RDN
163	STx Ø1
164	STx Ø2
165	RTN
166	LBL 15
167	"GANZE ZAHL= "
168	RTN
169	LBL 16
17Ø	"ZAEHLER= "
171	RTN
172	LBL 17
173	"NENNER= "
174	RTN
175	LBL 2Ø
176	ABS
177	STO Ø1
178	SF ØØ
179	RTN
18Ø	LBL 25
181	RCLFLAG
182	x<>y
183	FIX Ø
184	CF 29
185	ARCL X
186	x<>y
187	STOFLAG
188	x<>y
189	RTN
19Ø	END

14 Brüche, schnell gekürzt

Brüche kürzen ist nicht jedermanns Sache. Um diese lästige Tätigkeit so angenehm wie möglich zu machen, sollten Sie dieses Programm benützen.

14.1 Programmbedienung

Erklärung an Hand eines Beispieles:

Der Bruch $\frac{40318}{714}$ ist zu kürzen.

EINGABE	ANZEIGE
Start mit	
XEQ "BRUCH"	ZAEHLER=
40318 R/S	NENNER=
714 R/S	GGT= 2,0000
R/S	ZAEHLER= 20159,0000
R/S	NENNER= 357,0000

Der größte gemeinsame Teiler von 40318 und 714 ist 2, somit ist das Ergebnis $\frac{20159}{357}$.

verwendete Register : 1 bis 4

14.2 Anweisungsliste

Prog. Zeile	Befehl	Prog. Zeile	Befehl
1	LBL "BRUCH"	26	GTO ØØ
2	XEQ Ø2	27	LBL Ø1
3	PROMPT	28	RCL Ø3
4	STO Ø1	29	RCL Ø1
5	STO Ø3	3Ø	/
6	XEQ Ø3	31	STO Ø3
7	PROMPT	32	RCL Ø4
8	STO Ø2	33	RCL Ø1
9	STO Ø4	34	/
1Ø	LBL ØØ	35	STO Ø4
11	CLST	36	RCL Ø1
12	RCL Ø2	37	"GGT= "
13	x = y?	38	ARCL X
14	GTO Ø1	39	PROMPT
15	RCL Ø1	4Ø	LBL Ø4
16	RCL Ø2	41	RCL Ø3
17	/	42	XEQ Ø2
18	FRC	43	ARCL X
19	RCL Ø2	44	PROMPT
2Ø	x	45	RCL Ø4
21	,1	46	XEQ Ø3
22	+	47	ARCL X
23	INT	48	PROMPT
24	x<>Ø2	49	GTO Ø4
25	STO Ø1	5Ø	LBL Ø2

Prog. Zeile	Befehl
51	"ZAEHLER= "
52	RTN
53	LBL Ø3
54	"NENNER= "
55	RTN
56	END

Anhang

1 Datenspeicherung in einem File

FLREG
313 BYTES
= 63 ROWS

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

INPUT
171 BYTES
= 35 ROWS

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

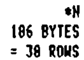
32

33

34

35

3 Fakultät bis 571 auf eine Stelle genau



1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

MIS
159 BYTES
= 32 ROWS

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

5 Division auf unendlich

DIV
85 BYTES
= 17 ROWS

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

6 Terminverwaltung

TERMIN
496 BYTES
= 100 ROWS

"**Die Programme** "TERMIN" und "TERMIN?" getrennt hintereinander einlesen und das eventu-**ell in der Mitte stehende** "END" löschen."

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

TERMIN?
394 BYTES
= 79 ROWS

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

7 Kuckucksuhr elektronisch

ZEITSIG
522 BYTES
= 105 ROWS

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

WECKER
168 BYTES
= 34 ROWS

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

9 Löschen des erweiterten Speichers

CLEM
26 BYTES
= 6 ROWS

1

2

3

4

5

6

CLEMX
41 BYTES
= 9 ROWS

1

2

3

4

5

6

7

8

9

10 Primfaktorzerlegung

PRIM
77 BYTES
= 16 ROWS

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

11 Quersummenberechnung

QSUM
50 BYTES
= 16 ROWS

1

2

3

4

5

6

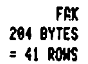
7

8

9

10

12 Fakultäten, Variationen und Kombinationen auch für Mengen größer 69



1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

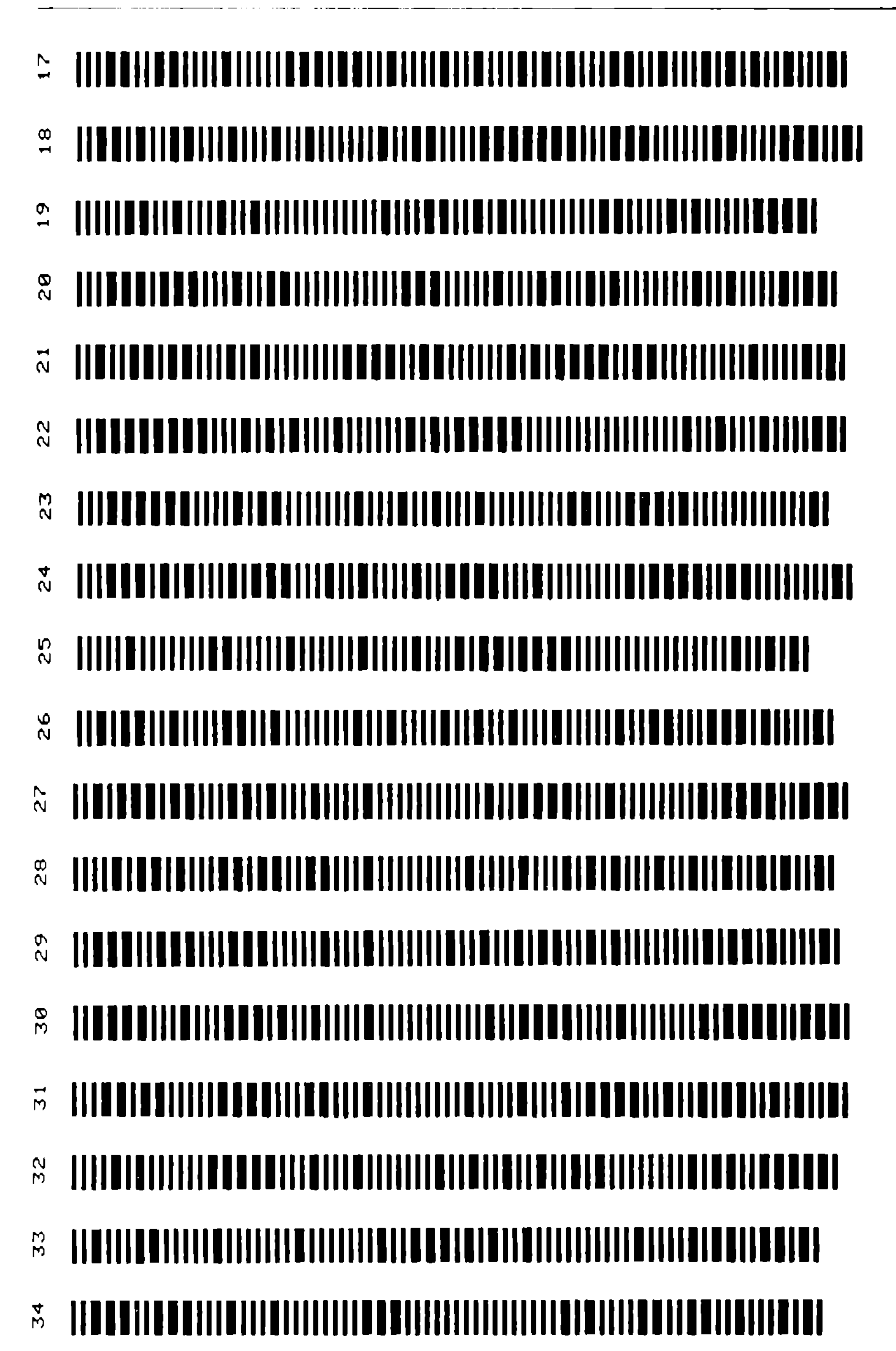
40

41

13 Bruchrechnen

BRE1
345 BYTES
= 69 ROWS

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16


17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

14 Brüche, schnell gekürzt

BRUCH
184 BYTES
= 21 ROWS

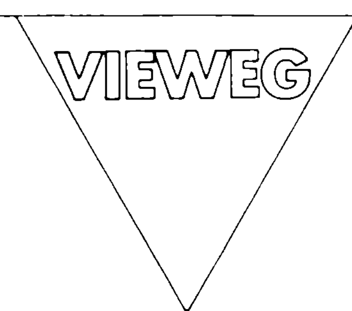
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17

18

19

20

21



Vieweg Programmbibliothek Mikrocomputer

Herausgegeben von Harald Schumny

Band 1
Graphik-Programme für TRS-80 und HP 9830

Band 2
Iterationen, Näherungsverfahren, Sortiermethoden
BASIC-Programme für CMB 3032, HP-9830, TRS-80, Olivetti 6060

Band 3
BASIC und Pascal im Vergleich

Band 4
BASIC-Anwenderprogramme

Band 5
BASIC-Programme für den PC-1211/1212

Band 6
Programme für den Einplatinencomputer TM 990/189
(16-Bit-Prozessor TMS 9980 A)

Band 7
PC-1500-Sammlung I

Band 8
Programme für den PC-1251

Band 9
PC-1500-Sammlung II

Band 10
PC-1500-Sammlung III

Band 11
Anwenderprogramme zum ZX-81 und ZX-Spectrum

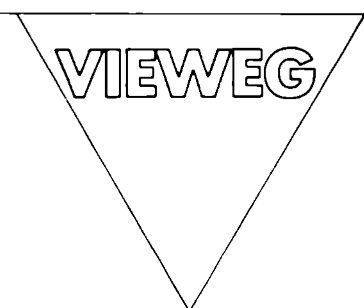
Band 12
17 Spiele für den PC-1500 A

Band 13
Ausgewählte BASIC-Computerspiele
(Atari 800)

Band 14
Lineares Optimieren
11 HP 41 Programme

Band 15
Dienstprogramme (Tool-Kit) für den HP-41

Band 16
Geodätische Berechnungsmethoden
(Standard-BASIC)



Vieweg Programmbibliothek Mikrocomputer

Herausgegeben von Harald Schumny

Band 17
Gelenkgetriebe für die Handhabungs- und Robotertechnik
(HP-41 CV)

Band 18
Probleme der Festigkeitslehre
(HP-41-Programme)

Band 19
PC-1500-Sammlung IV

Band 20
Dienstprogramme (Tool-Kit) für CBM 4032/8032

Band 21
HP-41 in der Praxis

Band 22
PC-1500-Sammlung V

Band 23
HP-41-Sammlung

Band 24
PC-1500-Sammlung VI

Band 25
Soziogramm mit dem Commodore 64

Band 26
Mathematikpaket für PC-1500 A

Band 27
Kryptologie-Programme
(HP-41 C/CV)

Band 28
HP-41 – Hilfen und Anwendungen

Band 29
BASIC-Programme zur Regelungstechnik (für den Sirius)

Band 30
Börse
Aktienanalyse mit dem Commodore 64

Band 31
PC-1500-Sammlung VII

Band 32
Programme aus Biologie, Chemie, Geographie mit dem Sharp PC-1211

Band 33
Mathematikprogramme für den Atari 800 XL

Band 34
PC-1500-Sammlung VIII

Band 35
Schachtrainer PC-1500 A

VIEWEG PROGRAMMBIBLIOTHEK
Mikrocomputer

Die Bände der Programmbibliothek enthalten ausgetestete Programme zu jeweils einem ausgewählten Themenschwerpunkt oder für einen aktuellen Mikrocomputer. Dabei wird der jeweilige Entwicklungsstand der Rechnertechnik berücksichtigt.

Die Programme sind, ausgehend von einer konkreten Aufgabenstellung, in der Regel in ihrem Ablauf beschrieben und durch ausgeführte Beispiele ergänzt. Wenn es nötig scheint, sind auch theoretische Grundlagen für die Programmierung erläutert.

Durch die graphischen, tabellarischen oder in Textform gegebenen Ablaufbeschreibungen wird die Übertragbarkeit auf andere Rechnertypen erleichtert, so daß die wirtschaftliche Nutzung der einzelnen Bände möglich ist. An Hand gleichartiger Aufgabenstellungen wird fallweise auch die unterschiedliche Arbeitsweise verschiedener Rechnertypen aufgezeigt.

Der Herausgeber bemüht sich ständig um eine sorgfältige Auswahl und Begutachtung der eingesandten Programme. Trotzdem kann keine Gewährleistung für vollständige Fehlerfreiheit übernommen werden. Programme zeigen ja oft erst nach vielen Testläufen mit wechselnden Parametern und Grenzbedingungen logische Fehlreaktionen und Sackgassen.

Für die Fälle, die zu Anregungen oder Kritik führen, sind in jedem Band die Anschriften der einzelnen Autoren angegeben. Wir erhoffen uns dadurch einen regen Gedankenaustausch zwischen Autoren und Benutzern der Programmbibliothek, der sich für beide Seiten als nützlich erweisen dürfte.

ISBN 3-528-04397-0